



TITLE:

# テスト網羅度関数の構築とソフトウェア信頼性評価への応用に関する考察 (不確実な状況における意思決定の理論と応用)

AUTHOR(S):

谷向, 和也; 井上, 真二; 山田, 茂

---

CITATION:

谷向, 和也 ...[et al]. テスト網羅度関数の構築とソフトウェア信頼性評価への応用に関する考察 (不確実な状況における意思決定の理論と応用). 数理解析研究所講究録 2008, 1589: 222-228

ISSUE DATE:

2008-04

URL:

<http://hdl.handle.net/2433/81566>

RIGHT:

## テスト網羅度関数の構築とソフトウェア信頼性評価への応用に関する考察

鳥取大学大学院・工学研究科 谷向 和也 (Kazuya Tanisaki)<sup>†</sup>

鳥取大学・工学部 井上 真二 (Shinji Inoue)<sup>††</sup>

鳥取大学・工学部 山田 茂 (Shigeru Yamada)<sup>††</sup>

<sup>†</sup>Graduate School of Engineering, Tottori University

<sup>††</sup>Faculty of Engineering, Tottori University

### 1 はじめに

近年、一般化枠組みに基づいてどのようなソフトウェア故障発生パターンにも比較的柔軟に対応できる SRGM の開発が行われている。しかし、それらは連続時間モデルに関するものが大半で、離散時間モデルに対しての議論はあまりなされていない。本研究では、離散時間上におけるソフトウェア信頼性モデルの一般化枠組みについて議論する。実際のテスト工程では、ソフトウェア故障発生現象に関するデータを採取するために、離散的にテスト期間中のソフトウェア故障発生頻度もしくは修正・除去されたフォールト数を計測する場合がほとんどである。離散時間 SRGM は、このような場合に対して比較的整合性を有するモデルとして考えることができる。また、実行されたテストケース数など離散値をとるテスト時間に依存したソフトウェア信頼度成長過程を記述する場合にも、同様のことが言える。

さらに、実際のテスト工程において計測可能なテスト網羅度 (testing-coverage) を取り上げ、テスト網羅度を考慮した離散型 SRGM を構築していく。テスト網羅度とは、ソフトウェアテストの十分性を評価するための指標として知られており、ソフトウェア開発のテスト工程における信頼度成長過程に影響を与える要因の 1 つとしても知られている。近年では、テスト工程におけるテスト網羅度達成状況の時間的推移とソフトウェア信頼度成長過程との関係性を考慮しながら、連続時間上におけるソフトウェア故障発生現象もしくはフォールト発見事象を、確率・統計則に基づいて記述する SRGM の開発がなされている [1]。

本研究では、実測データを用いた提案モデルの適用例を示し、既存の離散型 SRGM との適合性比較を行う。

### 2 離散型 SRGM 構築枠組み

本研究において構築する離散型 SRGM は、以下のような基本的仮定 [2] に基づいて議論される。

- (A1) 検出されたフォールトは、直ちにかつ完全に修正・除去される。
- (A2) 各フォールトは、それぞれ、独立かつ時間に関してランダムに検出かつ修正・除去され、検出されるまでの時間は、それぞれ、同一の離散型確率分布  $P(i) \equiv \Pr\{I \leq i\} = \sum_{k=0}^i p_I(k) (i = 0, 1, 2, \dots)$  に従う。ここで、 $p_I(k)$  および  $\Pr\{A\}$  は、それぞれ、 $I$  に関する確率関数および事象  $A$  に対する確率を表す。
- (A3) テスト開始前にソフトウェア内に潜在する総フォールト数 (初期潜在フォールト数)  $N_0$  は、ある確率分布に従う非負の確率変数とする。

いま、 $\{N(i), i = 0, 1, \dots\}$  をテスト開始後  $i$  期目までに発見される総フォールト数を表す離散型確率過程と定義する。上述の仮定 (A1) ~ (A3) から、テスト開始後  $i$  期目までに  $m$  個のフォールトが発見される確率関数は、

$$\Pr\{N(i) = m\} = \sum_n \binom{n}{m} \{P(i)\}^m \{1 - P(i)\}^{n-m} \Pr\{N_0 = n\} \quad (m = 0, 1, 2, \dots), \quad (1)$$

のように記述される。式 (1) において、フォールト発見事象に関する確率的挙動は、初期潜在フォールト数を表す非負の確率変数  $N_0$  に適切な確率関数を与えることで特徴付けられる。本研究では、式 (1) における  $N_0$  に対して、以下の仮定 [3] をおく。

- (B1) テスト開始時点におけるプログラムは、 $K$  コード行 (LOC) で構成される。

(B2) 各コードは、それぞれ一定の確率  $\lambda$  で 1 個のフォールトを含む。

(B3) プログラム内のコード中に潜在するフォールトは、それぞれ時間に関して独立かつランダムに検出される。

このとき、 $N_0$  は以下のようなパラメータ  $(K, \lambda)$  の二項分布を用いて表現できる。

$$\Pr\{N_0 = n\} = \binom{K}{n} \lambda^n (1 - \lambda)^{K-n} \quad (0 < \lambda < 1; n = 0, 1, \dots, K). \quad (2)$$

式 (1) および式 (2) から、テスト開始後  $i$  期目までに  $m$  個のフォールトが発見される確率関数は、次式のよ  
うに導出することができる [4]。

$$\Pr\{N_B(i) = m\} = \binom{K}{m} \{\lambda P(i)\}^m \{1 - \lambda P(i)\}^{K-m} \quad (0 < \lambda < 1; m = 0, 1, 2, \dots, K). \quad (3)$$

式 (3) より、フォールト検出時間分布  $P(i)$  に対して、ある適切な離散型確率分布を式 (3) に適用すること  
で、プログラム規模を考慮した離散型 SRGM を構築できる。

上述した離散型 SRGM の構築枠組みに基づいて、発見フォールト数の期待値と分散、ソフトウェア信頼  
度関数、瞬間 MTBF および累積 MTBF といったソフトウェア信頼性評価尺度を導出していく。このような  
ソフトウェア信頼性評価尺度は、ソフトウェア故障発生現象やフォールト発見事象を確率的に捉える SRGM  
に基づく信頼性評価を行う際に、ソフトウェア開発管理者に対して有益な情報を与える。

## 2.1 発見フォールト数の期待値と分散

式 (1) より、テスト開始後  $i$  期目までに発見される総フォールト数  $N(i)$  の期待値  $E[N(i)]$  と分散  $\text{Var}[N(i)]$   
は、以下のように導出することができる。

$$\begin{aligned} E[N(i)] &= \sum_{z=0}^n z \sum_n \binom{n}{z} \{P(i)\}^z \{1 - P(i)\}^{n-z} \Pr\{N_0 = n\} \\ &= E[N_0] P(i), \end{aligned} \quad (4)$$

$$\begin{aligned} \text{Var}[N(i)] &= E[N(i)^2] - (E[N(i)])^2 \\ &= \text{Var}[N_0] P(i)^2 + E[N_0] P(i) \{1 - P(i)\}. \end{aligned} \quad (5)$$

式 (4) および式 (5) から、式 (3) によって表される確率過程  $\{N_B(i), i = 0, 1, \dots\}$  の期待値および分散は、  
それぞれ、

$$E[N_B(i)] = K \lambda P(i), \quad (6)$$

$$\text{Var}[N_B(i)] = K \lambda P(i) \{1 - \lambda P(i)\}, \quad (7)$$

のように求められる。

## 2.2 ソフトウェア信頼度関数

ソフトウェア信頼度関数は、本研究で議論しているような離散型 SRGM を取り扱う場合、テスト開始後  
 $i$  期目までテストが進行しているときに、時間区間  $(i, i+h)$  ( $h = 0, 1, 2, \dots$ ) においてソフトウェア故障が  
発生しない確率と定義される。したがって、式 (1) から離散型ソフトウェア信頼度関数  $R(i, h)$  は、

$$\begin{aligned} R(i, h) &\equiv \sum_k \Pr\{N(i+h) = k \mid N(i) = k\} \cdot \Pr\{N(i) = k\} \\ &= \sum_k \left[ P(i+h)^k \{1 - P(i+h)\}^{-k} \cdot \sum_n \{1 - P(i+h)\}^n \binom{n}{k} \cdot \Pr\{N_0 = n\} \right], \end{aligned} \quad (8)$$

のように記述することができる。また、式 (8) から、初期潜在フォールト数  $N_0$  が式 (2) のような二項分布  
に従う場合、ソフトウェア信頼度関数  $R(i, h)$  は、

$$R(i, h) \equiv R_B(i, h) = [1 - \lambda \{P(i+h) - P(i)\}]^K, \quad (9)$$

のように求められる。

### 3 離散型テスト網羅度関数

2. において議論した基本的仮定の下でテスト網羅度達成状況に依存した離散型 SRGM を構築するために、テスト工程におけるテスト網羅度達成状況の時間的推移を考慮した離散型フォールト検出時間分布を考える。ここで、離散型フォールト検出時間分布に対するハザードレート (hazard rate) を、テスト開始後  $(i-1)$  期目までにフォールトが検出されないという条件の下で、その後の  $i$  期目においてフォールトが検出される確率として定義する。ハザードレートとは、一般的に連続時間上では、瞬間ソフトウェア故障発生確率であり、即時にソフトウェア故障が発生する割合、すなわちソフトウェア故障の発生度合いの瞬間的なスピードを意味し、信頼性工学において重要な信頼性尺度である。このとき、離散型ハザードレート  $z(i)$  は、

$$z(i) = \Pr\{I = i \mid I > i-1\} = \frac{P(i) - P(i-1)}{1 - P(i-1)}, \quad (10)$$

と表現できる。式 (10) より、離散型フォールト検出時間分布  $P(i)$  は、以下のように求められる。

$$\left. \begin{aligned} P(0) &= 0 \\ P(i) &= 1 - \prod_{j=1}^i (1 - z(j)) \quad (i = 1, 2, \dots) \end{aligned} \right\}. \quad (11)$$

式 (11) から、離散型フォールト検出時間分布を与えることは、それに対応した離散型ハザードレートを与えるということになる。

本研究では、テスト網羅度達成状況の時間的推移がソフトウェア信頼度成長過程に与える影響を考慮した離散型 SRGM を構築するにあたり、式 (10) の離散型ハザードレートを次式のように与える。

$$\begin{aligned} z(i) &= \phi(i)\{C(i) - C(i-1)\} \\ &= \phi(i)c(i). \end{aligned} \quad (12)$$

式 (12) において、 $\phi(i)$  はテスト開始後  $i$  期目における単位テスト網羅度当りのハザードレートであり、 $C(i)$  はテスト開始後  $i$  期目までに達成されたテスト網羅度、 $c(i)$  はテスト開始後  $i$  期目において達成されたテスト網羅度を表す関数である。特に、 $C(i)$  をテスト網羅度関数 (testing-coverage function) と呼ぶこととする。

### 4 離散型テスト網羅度関数

本研究では、式 (12) におけるテスト網羅度関数  $C(i)$  に関して、以下に述べる 2 種類のモデルを構築する。

#### 4.1 $\alpha$ テスト網羅度モデル

テスト網羅性尺度の中でも、テスト工程においてテスト網羅度 100% を達成することが現実的な意味で困難である尺度 (例えば、パス網羅尺度) に対して適用すべきモデルである。本研究では、テストケース設計者の設計能力を考慮した、以下に示すモデルを用いる。

$$C(i) = \frac{\alpha[1 - \{(1 - \frac{1}{2}b)/(1 + \frac{1}{2}b)\}^i]}{1 + s\{(1 - \frac{1}{2}b)/(1 + \frac{1}{2}b)\}^i} \quad (0 < \alpha < 1, s > 0, b > 0). \quad (13)$$

上式は、テストケース設計者の設計能力を考慮した連続時間テスト網羅度関数 [1] の基本的仮定から導出されるリカッチ型微分方程式を、可積分な差分手法 [5] を用いて導出された以下の可積分な差分方程式

$$C_{n+1} - C_n = rab + \frac{b(1-2r)}{2}[C_n + C_{n+1}] - \frac{b(1-r)}{\alpha}C_n C_{n+1}, \quad (14)$$

の厳密解である。式 (13)、式 (14) において、 $\alpha$  はテスト終了時におけるテスト網羅度の達成目標値、 $b$  は単位テスト期数当りのテスト網羅度達成率、 $s$  はテストケース設計者の設計能力を表すパラメータである。式 (13) を式 (12) における離散型テスト網羅度関数として用いた離散型 SRGM を、「Model 1」と呼ぶことにする。

## 4.2 [0,1] 網羅度モデル

ある一定のソフトウェア品質や信頼性を保証するために、テスト工程において 100% のテスト網羅度を達成しなければならない、もしくは、100% のテスト網羅度が現実的に達成可能なテスト網羅性尺度 (例えば、命令網羅尺度) に対して適用すべきモデルである。本研究では、テスト網羅度がとり得る値を考慮して、テスト網羅度関数  $C(i) (i = 0, 1, 2, \dots)$  のロジット変換 (logit transform) [6] 値が、それぞれ、次に示す回帰式によって与えられる場合を考える。

$$L(C(i)) \equiv \ln \frac{C(i)}{1-C(i)} = \begin{cases} \beta_0 + \beta_1 i \\ \beta_0 + \beta_1 i + \beta_2 i^2. \end{cases} \quad (15)$$

ここで、パラメータ  $\beta_0$  は定数であり、 $\beta_1$  および  $\beta_2$  は、それぞれ、回帰係数を表す。式 (15) から、ロジット逆変換を行うとテスト網羅度関数は、それぞれ、

$$\begin{aligned} C(i) &= L^{-1}\{C(i)\} \\ &= \begin{cases} \frac{1}{1+\exp[-\beta_0-\beta_1 i]} \\ \frac{1}{1+\exp[-\beta_0-\beta_1 i-\beta_2 i^2]}, \end{cases} \end{aligned} \quad (16)$$

のように導出される。式 (16) において、テスト網羅度関数のロジット変換がテスト期数に関する 1 次の回帰式で表された場合のテスト網羅度関数を用いた離散型 SRGM を [Model 2]、2 次の回帰式で表されたものを用いた離散型 SRGM を [Model 3] と呼ぶことにする。

## 5 パラメータ推定

一定のテスト時間区間  $(0, t_k]$  において達成されたテスト網羅度およびそれに伴い発見された総フォールト数  $y_k$  に関する  $N$  組のフォールト発見数データ  $(t_i, y_k, c_k) (k = 0, 1, 2, \dots, N; t_0 < t_1 < t_2 < \dots < t_N; t_0 = 0; y_0 = 0; c_0 = 0)$  が観測されたものとする。本研究において提案した離散型 SRGM のパラメータ推定手法について議論する。まず、式 (13) のテスト網羅度関数に含まれるパラメータは、最小 2 乗法を用いて推定できる。前述した式 (14) から、以下の線形式を導出する。

$$Y_n = A + B_1 K_n + B_2 L_n. \quad (17)$$

ここで、式 (17) において、

$$\begin{cases} Y_n = C_{n+1} - C_n \\ K_n = C_n + C_{n+1} \\ L_n = C_n C_{n+1} \\ A = \alpha r b \\ B_1 = b(1 - 2r)/2 \\ B_2 = -b(1 - r)/\alpha, \end{cases} \quad (18)$$

である。式 (17) より、回帰係数  $A, B_1$  および  $B_2$  の推定値、 $\hat{A}, \hat{B}_1$  および  $\hat{B}_2$  は、観測されたテスト網羅度データから推定できる。したがって、式 (18) より、パラメータ  $\alpha, b, r$  および  $s$  の推定値  $\hat{\alpha}, \hat{b}, \hat{r}$  および  $\hat{s}$  は、それぞれ、

$$\begin{cases} \hat{\alpha} = \hat{A}/(\sqrt{\hat{B}_1^2 - \hat{A}\hat{B}_2} - \hat{B}_1) \\ \hat{b} = 2\sqrt{\hat{B}_1^2 - \hat{A}\hat{B}_2} \\ \hat{r} = (1 - \hat{B}_1/\sqrt{\hat{B}_1^2 - \hat{A}\hat{B}_2})/2 \\ \hat{s} = (1 - \hat{r})/\hat{r}, \end{cases} \quad (19)$$

と推定できる。

式 (16) に示したテスト網羅度関数に対するパラメータ推定手法は、式 (15) に与えた回帰式に基づいて、通常よく用いられる最小 2 乗法を適用し、パラメータ  $\beta_0, \beta_1$ , および  $\beta_2$  の推定値,  $\hat{\beta}_0, \hat{\beta}_1$ , および  $\hat{\beta}_2$  を推定することになる。

次に、テスト網羅度関数が推定された下で、パラメータ  $(\lambda, \phi)$  を推定する方法について議論する。テスト  $t_i (i = 0, 1, 2, \dots, N)$  期目までに発見される総フォールト数  $N_B(t_i)$  に関する尤度関数  $l$  は、

$$\begin{aligned} l &\equiv \Pr\{N_B(t_1) = y_1, N_B(t_2) = y_2, \dots, N_B(t_N) = y_N\} \\ &= \prod_{i=2}^N \Pr\{N_B(t_i) = y_i \mid N_B(t_{i-1}) = y_{i-1}\} \cdot \Pr\{N_B(t_1) = y_1\} \end{aligned} \quad (20)$$

と表される。式 (20) における条件付確率  $\Pr\{N_B(t_i) = y_i \mid N_B(t_{i-1}) = y_{i-1}\}$  は、

$$\begin{aligned} &\Pr\{N_B(t_i) = y_i \mid N_B(t_{i-1}) = y_{i-1}\} \\ &= \binom{K - y_{i-1}}{y_i - y_{i-1}} \{z(t_{i-1}, t_i)\}^{y_i - y_{i-1}} \{1 - z(t_{i-1}, t_i)\}^{K - y_i} \end{aligned} \quad (21)$$

のように求められる。ここで、

$$z(t_{i-1}, t_i) = \frac{\lambda\{P(t_i) - P(t_{i-1})\}}{1 - \lambda P(t_{i-1})}, \quad (22)$$

である。したがって、式 (20) は、式 (21) を用いて最終的に次のように記述できる。

$$l = \prod_{i=1}^N \binom{K - y_{i-1}}{y_i - y_{i-1}} \{z(t_{i-1}, t_i)\}^{y_i - y_{i-1}} \{1 - z(t_{i-1}, t_i)\}^{K - y_i}. \quad (23)$$

但し、 $P(0) = 0$  である。式 (23) から、対数尤度関数は、

$$\begin{aligned} L &\equiv \log l \\ &= \log K! - \log\{(K - y_N)!\} - \sum_{i=1}^N \log\{(y_i - y_{i-1})!\} + y_N \log \lambda \\ &\quad + \sum_{i=1}^N (y_i - y_{i-1}) \log\{P(t_i) - P(t_{i-1})\} + (K - y_N) \log\{1 - \lambda P(t_N)\} \end{aligned} \quad (24)$$

のように求められる。ここで、 $P(i)$  が式 (11) および式 (12) から導出された

$$\begin{cases} P(0) = 0 \\ P(i) = 1 - \prod_{j=1}^i (1 - \phi(j)C(j)) \end{cases} \quad (25)$$

に従うと仮定した場合、対数尤度関数は式 (24) より、

$$\begin{aligned} L &= \log K! - \log\{(K - y_N)!\} - \sum_{i=1}^N \log\{(y_i - y_{i-1})!\} + y_N \log \lambda \\ &\quad + \sum_{i=1}^N (y_i - y_{i-1}) \log\left\{\prod_{j=1}^{i-1} (1 - \phi(j)C(j)) - \prod_{j=1}^i (1 - \phi(j)C(j))\right\} \\ &\quad + (K - y_N) \log\left\{1 - \lambda\left(1 - \prod_{j=1}^N (1 - \phi(j)C(j))\right)\right\} \end{aligned} \quad (26)$$

と導出される。これより、

$$\frac{\partial L}{\partial \lambda} = \frac{\partial L}{\partial \phi} = 0, \quad (27)$$

となる同時尤度方程式を数値的に解くことにより、パラメータ  $\lambda$  および  $\phi$  の最尤推定値  $\hat{\lambda}$  および  $\hat{\phi}$  が得られる。

表 1: MSE に基づいたモデルの適合性比較結果.

	DS1	DS2
Model 1	80.502	—
Model 2	<b>28.316</b>	<b>90.796</b>
Model 3	—	—
D-GOMP	70.344	744.89
D-LOGI	346.78	1103.6

## 6 モデルの適合性比較

本研究では、2組の実測データ (DS1 および DS2 と呼ぶことにする) を用い、平均偏差 2 乗和 (MSE) の観点から、本研究で提案した離散型 SRGM と既存の離散型 SRGM との適合性比較を行う。既存の離散型 SRGM として、離散型ゴンプertz 曲線モデル (discrete Gompertz curve model: D-GOMP) および離散型ロジステック 曲線モデル (discrete logistic curve model: D-LOGI) [7] を取り上げる。

表 1 に、MSE に基づいたモデルの適合性比較結果を示す。MSE の値が小さいほど、推定されたモデルの実測データに対する適合性が高いことが言える。今回は、(Model 3) に関して推定することが出来なかった。DS1 の場合、既存のモデルで一番適合性の高かった D-GOMP と比べて (Model 2) の適合性が高く、全体的にも (Model 2) の適合性が一番高い結果となった。DS2 では、提案したモデルの中でも (Model 2) 以外の結果が出なかったが、既存モデルである D-GOMP や D-LOGI よりも高い適合性を示した。

以上の結果から、今回提案した離散型 SRGM は、本研究で取り上げた既存の離散型 SRGM と比較して、実測データに対する適合性が向上していることが伺える。しかしながら、適合性を比較するために用いた実測データは 2 種類であったため、さらに提案モデルに適用可能な実測データを収集して、より多くの実測データを用いた適合性比較とテスト網羅度関数の妥当性の評価を行う必要がある。

## 7 ソフトウェア信頼性解析例

表 1 より、精度の高かった (Model 2) の数値例を取り上げる。図 1 に DS1 を用いて推定された発見フォールト数の期待値およびその 95% 信頼限界を、図 2 に DS2 を用いて推定された発見フォールト数の期待値およびその 95% 信頼限界をそれぞれ示す。また、図 3 に DS1 を用いたテスト期数  $i = 24$  以降におけるソフトウェア信頼度の挙動を、図 4 に DS2 を用いたテスト期数  $i = 24$  以降におけるソフトウェア信頼度の挙動を示す。

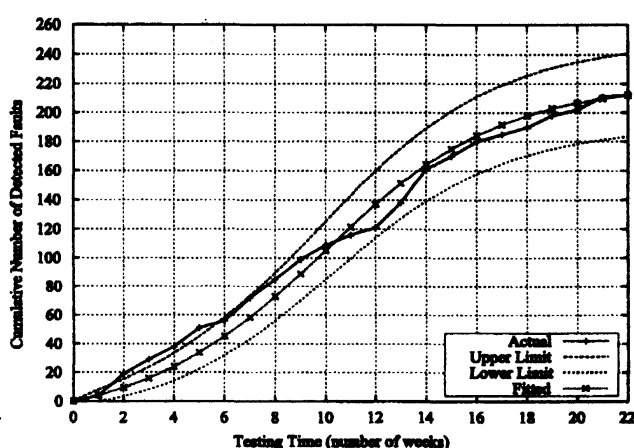
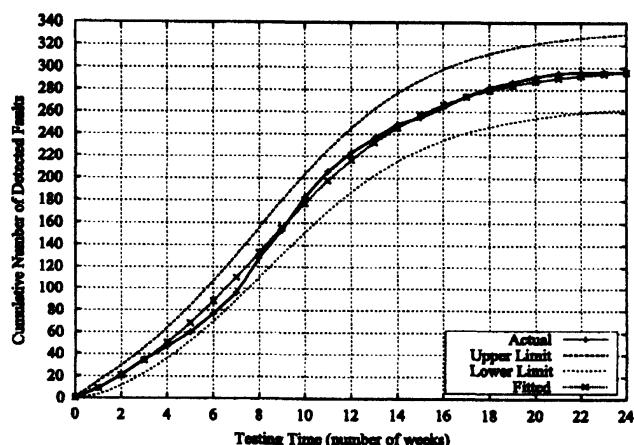


図 1: 推定された発見フォールト数の期待値およびその 95% 信頼限界 (Model 2; DS1).  
 図 2: 推定された発見フォールト数の期待値およびその 95% 信頼限界 (Model 2; DS2).

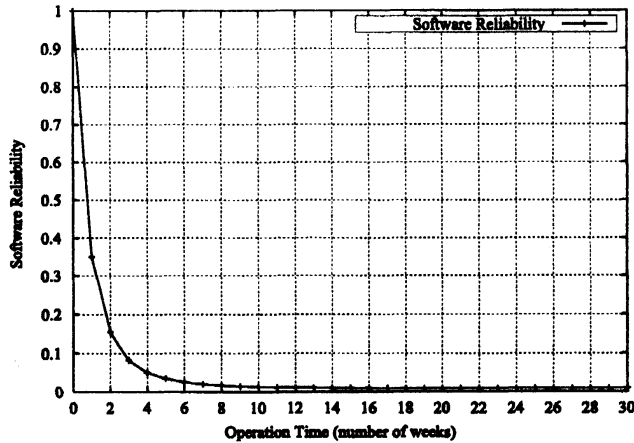
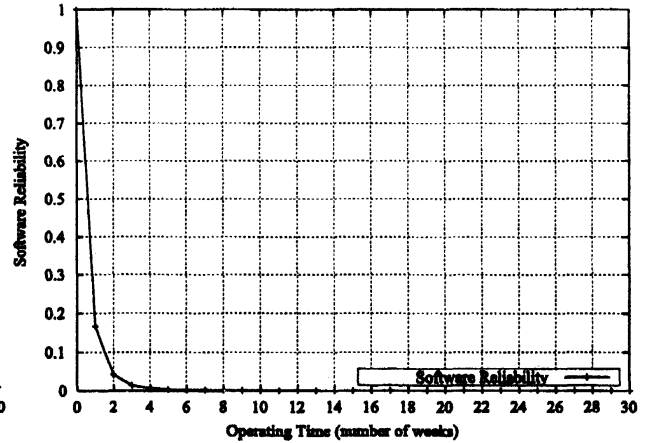


図 3: 推定されたソフトウェア信頼度関数, 図 4: 推定されたソフトウェア信頼度関数,  $\hat{R}_B(24, h)(\text{Model 2;DS1})$ .



$\hat{R}_B(24, h)(\text{Model 2;DS2})$ .

## 8 むすび

本研究では、離散型 SRGM の一般化枠組みに基づいて、初期潜在フォールト数に対する確率分布がパラメータ  $K$  および  $\lambda$  をもつ二項分布に従う場合における離散型 SRGM の構築枠組みについて議論した。また、テスト網羅度依存型 SRGM を構築するために、テスト網羅度を考慮した離散型ハザードレートとそれに対する離散型フォールト検出時間分布を求めた。さらに、テスト網羅度関数に関して、 $\alpha$  テスト網羅度モデル、 $[0,1]$  網羅度モデルと名付けた 2 種類のモデルを構築した。本研究で提案した離散型 SRGM と既存の離散型 SRGM との適合性を比較するために、実測データを用いたモデルの適合性比較も行った。適合性比較の結果、既存の離散型 SRGM に比べ MSE の観点から、提案した離散型 SRGM の実測データに対する適合性が向上しているということが確認できた。今後は、本研究で提案した離散型 SRGM に適用可能な実測データを収集し、それらを用いた適合性の比較と、テスト網羅度関数に対する妥当性の評価を行う必要がある。

## 謝辞

本研究の一部は、日本学術振興会科学研究費補助金 若手研究 (B) (課題番号 19710129)、基盤研究 (C) (課題番号 18510124)、および鳥取大学ベンチャー・ビジネス・ラボラトリー平成 19 年度提案型研究開発テーマの援助を受けたことを付記する。

## 参考文献

- [1] S. Inoue and S. Yamada, "Testing-coverage dependent software reliability growth modeling, " *Intern. J. Reliab. Quali. Safe. Eng.*, Vol. 11, No. 4, pp. 303–312, 2004.
- [2] H. Okamura, A. Murayama, and T. Dohi, "EM algorithm for discrete software reliability models: a unified parameter estimation method, " *Proc. 8th IEEE Intern. Symp. High Assura. Syst. Eng. (HASE'04)*, 2004, pp. 219–228.
- [3] M. Kimura, S. Yamada, H. Tanaka, and S. Osaki, "Software reliability measurement with prior information on initial fault content, " *Trans. IPS Japan*, Vol. 34, No. 7, pp. 1601–1609, 1993.
- [4] S. Inoue and S. Yamada, "Generalized discrete software reliability modeling with effect of program size, " *IEEE Trans. Syst. Man. Cybern. -Part A: Syst. Hum.*, Vol. 37, No. 2, pp. 170–179, March 2007.
- [5] R. Hirota, "Nonlinear partial difference equations. V. Nonlinear equations reducible to linear equations, " *Journal of the Physical Society of Japan*, Vol. 46, No. 1, pp. 312–319, January 1979.
- [6] 圓川隆夫, 宮川雅巳, 「SQC—理論と実際—」, 朝倉書店, 東京, 1992.
- [7] D. Satoh and S. Yamada, "Discrete equations and software reliability growth models, " *Proc. 12th IEEE Intern. Symp. Softw. Reliab. Eng. (ISSRE'01)*, pp. 176–184, 2001.